REMARKS

Claims 1, 2, 4-37, 40-55, 59, 63-91, 95, 99-114, and 118-147 are pending after this amendment.

Applicants have amended claims 1, 2, 13, 18, 20, 23, 31, 37, 53, 59, 63, 64, 82, 95, 102, 111, and 118 in order to more particularly define the invention. The amendments were not necessitated by the claim rejections. Applicants make no admission as to the patentability or unpatentability of the originally filed claims.

Claims 3, 38, 39, 56-58, 60-62, 92-94, 96-98, and 115-117 have been canceled.

Claims 120-147 have been added.

The amendments and remarks presented herein are in response to the Office Action dated June 7, 2004.

The Examiner objected to claims 20 and 31 because of informalities related to typographical errors. The errors have been corrected.

The Examiner rejected claims 1-5, 9-17, 21-25, 27-40, 42-48, 50-54, 56-58, 66, 69, 73-76, 78-87, 89-94, 102-105, 107-113, and 115-117 under 35 USC 102(b) as being anticipated by Grover et al. This rejection is respectfully traversed.

Claim 1, which has been amended merely to clarify the subject matter of the invention, recites:

> A computer-implemented method for concurrently accepting parameters in at least two contexts, the method comprising:

accepting a keystroke sequence comprising at least one keystroke, each keystroke having a first value, and at least a subset of the keystrokes having a second value;
determining whether the keystroke sequence produces a valid result in a first context;
responsive to the keystroke sequence producing a valid result in the first context, outputting first feedback, the first feedback indicating keystroke input according to the first context; and
responsive to the keystroke sequence not producing a valid result in the first context:
determining whether the keystroke sequence produces a valid result in a second context; and
responsive to the keystroke sequence producing a valid result in the second context, outputting second feedback, the second feedback indicating keystroke input according to the second context.

The claimed method accepts a keystroke sequence and performs one of two different actions depending on whether or not the sequence produces a valid result in a first context. Specifically, if the sequence does produce a valid result in the first context, it outputs first feedback indicating the keystroke input according to the first context. Alternatively, if the sequence does not produce a valid result in the first context, it performs the following two substeps: first it determines whether the keystroke sequence produces a valid result in a second context, and then, if the keystroke sequence does produce a valid result in the second context, it outputs second feedback indicating keystroke input according to the second context.

A particular advantage of this method is that the two conditions are mutually exclusive: first feedback is output in response to a valid result in the first context, and second feedback is output in response to no valid result in the first context (and a valid results in the second context). By definition, these two conditions cannot occur simultaneously. This avoids confusion on the part of the user, and further avoids the need for the user to select from two or more simultaneously displayed interpreta-

tions of the keystroke sequence in different contexts. Rather, the method selects, on behalf of the user, a first (preferred) context when there is a valid result in that context, and resorts to the second context when there is no valid result in the first context. The result is an elegant, simple solution for accepting user input that can be interpreted in two or more ways.

An example of an implementation of this method as applied to directory search and direct-dial input functions is described in the specification at paragraph 0067. In this example, the invention assumes that the user is attempting to perform directory filtering (the first context), and displays directory filtering results accordingly. As long as the directory filter produces at least one valid result, the filter results continue to be displayed. If, however, the directory filter produces no results, and the keystroke sequence has a numeric value, then the numeric value of the keystroke sequence (the second context) is displayed instead of directory filtering results. Thus, the invention switches to displaying dialed numbers if it becomes evident that the user is performing a direct-dial operation.

It is important to note that the displayed filtering results can include any number of individual directory records. For example, the entered string "Andr" might match "Andrew Smith", "Richard Andrews", and "Andrea Jones", so that all three of these names might be displayed. Thus, the claimed invention does not preclude the possibility of requiring the user to select among multiple results within one of the contexts.

Although this embodiment is merely an example, it serves to illustrate the usefulness of the claimed invention. The user need only enter the keystroke sequence, and the invention automatically determines whether the user it attempting to provide input in the first context (such as directory filtering) or the second context (such as direct-dial). The invention is particularly useful in situations such as that described in the embodiment, where the likelihood that wrong context will be inferred is extremely small, owing to the relatively limited set of valid results in the first context (a valid result in the first context for this example corresponds to a result that matches a name in a limited set of names, i.e., a stored directory database).

By contrast, Grover merely describes techniques for implementing a reduced keyboard wherein a plurality of letters and symbols are assigned to most of the keys. When more than one word matches an entered keystroke sequence, Grover presents a menu from which the user selects the desired word. For example, as shown in Fig. 7g of Grover, where the user has entered a three-stroke sequence, the display shows three possible matching choices in an on-screen menu; the user then manually selects the desired word from the menu.

Nowhere in Grover is there any mention of a method where first feedback corresponding to a first context is shown when there is a valid result in the first context, and second feedback corresponding to a second context is shown responsive to there being no valid result in the first context, as claimed herein. In fact, given the open-ended word lookup methodology described in Grover, such a methodology would not be effective in the context of Grover. Rather, in Grover it is necessary that the

user be given an opportunity to select from two or more choices, since there are many cases where more than one match is plausible.

Thus, not only does Grover fail to describe the particular method recited in claim 1, but also the described methodology of Grover addresses a different problem than that of the invention claimed herein, and therefore fails to hint or suggest the claimed invention.

Claims 2, 4-5, 9-17, 21-25, and 27-36 depend from amended claim 1, and therefore incorporate all of the limitations of amended claim 1. The arguments presented above therefore apply to these claims, which are distinguishable from Grover for the reasons stated above.

Furthermore, these claims recite additional limitations that further distinguish them from Grover. For example, claim 2, which has been amended to clarify the subject matter of the invention and to incorporate the limitations of claim 3, as amended recites, in part, "responsive to the keystroke sequence producing a valid result in the first context, performing a first action corresponding to the first context, using the first value for each keystroke" and "responsive to the keystroke sequence not producing a valid result in the first context and producing a valid result in the second context, performing a second action corresponding to the second context, using the second value for each keystroke." Claim 36 recites, in part, "responsive to the keystroke sequence not producing a valid result in one of the contexts, performing an action using the keystroke sequence according to the other context."

For example, if the keystroke sequence produces a valid result in a directory lookup context, the invention filters the directory using the first value for each keystroke; conversely, if the keystroke sequence does not produce a valid result in the directory lookup context but produces a valid result in a direct-dialing context, the invention direct-dials using the second value for each keystroke. Thus, the invention automatically selects an action to perform depending on which context produces a valid result. These actions are in addition to outputting feedback indicating keystroke input.

Grover does not teach or suggest any such technique. Rather, as acknowledged by the Examiner, Grover merely describes displaying different words that match different interpretations of the keystroke sequence. There is no mention of any actions that are taken, in addition to outputting feedback, in response to whether or not valid results exist in different contexts.

As another example, claim 23, which has been amended merely to clarify the subject matter of the invention, recites:

> "The method of claim 1, further comprising:
> accepting a backspace keystroke;
> deleting a keystroke from the keystroke sequence;
> repeating the steps of:
> determining whether the keystroke sequence produces a valid result in a first context;
> responsive to the keystroke sequence producing a valid result in the first context, outputting first feedback, the first feedback indicating keystroke input according to the first context; and
> responsive to the keystroke sequence not producing a valid result in the first context:
> determining whether the keystroke sequence produces a valid result in a second context; and

> responsive to the keystroke sequence producing a valid result in the second context, outputting second feedback, the second feedback indicating keystroke input according to the second context."

The user can therefore backtrack by activating a backspace keystroke. In response, the method deletes a keystroke from the sequence and repeats the steps to determine what type of feedback to provide.

Grover fails to describe any such technique. The Delete key (reference character 106) of Grover is used for deleting entire words, not for deleting individual keystrokes as claimed herein. At col. 8, lines 45-50 Grover describes the well-known technique of assigning on-screen buttons to different types of functions; specifically, it states that "system keys 104-106 can be programmed to directly perform a system function (for example, delete the last word entered) or to present a system menu (for example, present a menu of additional delete options)." (Emphasis added).

As another example, claim 27 recites, in part, that "determining whether the keystroke sequence produces a valid result in a first context comprises determining whether all of the accepted keystrokes have a numeric value." Thus, the claimed method determines whether a valid result exists in a first context (such as for example direct entry of a telephone number) by determining whether all of the keystrokes entered thus far have a numeric value. If at least one of the keystrokes does not have a numeric value, it can be assumed that the user is not attempting direct entry of a telephone number (or some other numeric entry). Thus, the claimed method pro-

vides a technique for determining the user's intention from the input keystroke sequence alone, without requiring the user to explicitly indicate his intention.

The Examiner states that Grover teaches these steps because it describes interpreting the accepted keystrokes of "AFG", "DHI", "DHI" as entry of numeric digits "1", "6", "6". It is acknowledged that the keys in Grover have both alphabetic and numeric values. However, Grover fails to teach any technique for determining whether a sequence produces a valid result in a particular context by determining whether the entered keystrokes all have a numeric value, as claimed herein.

In fact, such a step would make no sense in the context of Grover, since all of the character entry keys have a numeric value. As can be seen from Fig. 1, there are ten character entry keys, each of which has a numeric value. Thus, it would make no sense to determine whether a series of entered keystrokes all have a numeric value, since every keystroke sequence entered via the input method of Grover would satisfy this condition. In other words, Grover does not teach any mechanism determining whether or not the user intends to enter a numeric sequence, since every character entry key has a numeric value.

A similar argument applies with respect to claim 32, which recites in part that "determining whether the keystroke sequence produces a valid result in a second context comprises determining whether all of the accepted keystrokes have a numeric value."

As another example, claim 35 recites, in part, "responsive to at least one of the accepted keystrokes not being valid in one of the contexts, determining that the other context is intended." The claimed method thus makes a determination as to which context the user intended. The determination is made based on detection of at least one keystroke that is not valid in one of the contexts; responsive to such an event, the other context is inferred.

As stated above, Grover makes no mention of determining which of a plurality of contexts is intended. Rather Grover merely presents all valid possibilities and allows the user to select among them. The Examiner stated that the claimed step is taught by Grover at col. 4, lines 61-64, col. 14, lines 15-26, and Fig. 1. However, these portions of Grover merely disclose the fact that keystrokes can be interpreted as digits as well as being interpreted as alphabetic characters. Again, there is no discussion of any technique of determining an intended context. In fact, other than issuing a tone to indicate that the selection list is empty, there is no discussion in Grover of any action being taken (or determination being made) in response to invalidity of keystroke in a given context.

Claim 3 has been canceled.

Claim 37 has been amended to incorporate the limitations of claims 38 and 39. Claim 37 as amended recites:

"A computer-implemented method for concurrently accepting parameters in at least two contexts, the method comprising:

accepting a keystroke sequence comprising at least one keystroke, each keystroke having a first value, and at least a subset of the keystrokes having a second value;

determining whether the keystroke sequence produces a valid result in a first context;

responsive to the keystroke sequence producing a valid result in the first context:

outputting first feedback, the first feedback indicating keystroke input according to the first context; and

performing a first action corresponding to the first context, using the first value for each keystroke;

determining whether the keystroke sequence produces a valid result in a second context; and

responsive to the keystroke sequence producing a valid result in the second context:

outputting second feedback, the second feedback indicating keystroke input according to the second context; and

performing a second action corresponding to the second context, using the second value for each keystroke."

Thus, when a keystroke sequence produces a valid result in the first context, first feedback is output and a first action is performed. Conversely, when a keystroke sequence produces a valid result in the second context, second feedback is output and a second action is performed. The user need not explicitly specify the action to be performed, as the claimed method infers the action to be taken based on which valid results exist and in which context. Rather, the invention automatically selects an action to perform depending on which context produces a valid result. For example, if the keystroke sequence produces a valid result in a directory lookup context, the invention filters the directory using the first value for each keystroke; conversely, if the keystroke sequence produces a valid result in a direct-dialing context, the invention direct-dials using the second value for each keystroke. These actions are in addition to outputting feedback indicating keystroke input.

Grover does not teach or suggest any such technique. Rather, as acknowledged by the Examiner, Grover merely describes displaying different words that match different interpretations of the keystroke sequence. There is no mention of any actions that are taken, in addition to outputting feedback, in response to whether or not valid results exist in different contexts.

Claims 40, 42-48, and 50-51 depend from amended claim 37, and therefore incorporate all of the limitations of amended claim 37. The arguments presented above therefore apply to these claims, which are distinguishable from Grover for the reasons stated above.

Furthermore, these claims recite additional limitations that further distinguish them from Grover.

For example, claim 47 recites:

> "The method of claim 37, further comprising:
> accepting a backspace keystroke;
> deleting a keystroke from the keystroke sequence;
> repeating the steps of determining whether the keystroke sequence produces a valid result in a first context, and, responsive to the keystroke sequence producing a valid result in the first context, outputting first feedback, the first feedback indicating keystroke input according to the first context; and
> repeating the steps of determining whether the keystroke sequence produces a valid result in a second context, and, responsive to the keystroke sequence producing a valid result in the second context, outputting second feedback concurrently with the first feedback, the second feedback indicating keystroke input according to the second context."

The arguments presented above in connection with claim 23 apply to claim 47.

As another example, claim 50 recites, in part, that "determining whether the keystroke sequence produces a valid result in a first context comprises determining

whether all of the accepted keystrokes have a numeric value." The arguments presented above in connection with claim 27 apply to claim 50.

Claims 38-39 have been canceled.

Claim 52 recites:

> "A computer-implemented method for concurrently accepting parameters in at least two contexts, the method comprising:
>
> a) initiating a first string;
> b) accepting a keystroke;
> c) appending a first value of the keystroke to the first string;
> d) determining whether all values in the first string can be converted to valid numeric values;
> e) responsive to determining that all values in the first string can be converted to valid numeric values:
> generating a numeric string corresponding to the first string; and
> outputting first feedback comprising the numeric string;
> f) determining whether any directory records match the first string; and
> g) responsive to at least one directory record matching the first string, outputting second feedback comprising a list of the at least one directory record matching the first string."

In steps d) and e), the method determines whether all values can be converted to valid numeric values and outputs first feedback in response to a positive determination. This automatic determination of whether a numeric value exists for each keystroke is useful because it avoids the need for the user to manually or explicitly specify whether a numeric entry is intended.

Grover teaches no such step. In fact, as stated above, in Grover's device every character entry key includes a numeric value. Thus, it would make no sense to determine whether a series of entered keystrokes all have numeric values, since every keystroke sequence entered via the input method of Grover would satisfy this condi-

tion. In other words, Grover does not teach any mechanism determining whether or not the user intends to enter a numeric sequence, since every character entry key has a numeric value.

Claims 53-54 depend from claim 52, and therefore incorporate all of the limitations of claim 52. The arguments presented above therefore apply to these claims, which are distinguishable from Grover for the reasons stated above.

Furthermore, these claims recite additional limitations that further distinguish them from Grover. For example, claim 54 recites, in part, "responsive to determining that at least one value in the first string cannot be converted to a valid numeric value, deleting any previously output first feedback comprising the numeric string." Thus, the invention deletes previously output numeric string feedback in the event that at least one value in the string cannot be converted to a valid numeric value. This serves to automatically update the display based on what valid interpretations exist for the input string.

Grover fails to teach any such step. In fact, in Grover all character input keys have numeric values, so that there would never be a case where one of the values in an input string could not be converted to a valid numeric value. In other words, in Grover, every input character can be converted to a valid numeric value; therefore the recited step would never be performed, nor would there be a need to perform it.

The Examiner cites the use of the Delete key as an example. However, Grover's Delete key is a key that performs a specific function (deleting the most recently entered word, for example) rather than providing a value in a string. In

Grover, hitting the Delete key would never result in a string value that cannot be converted to a valid numeric value.

Claims 56-58 have been canceled.

Claims 66, 69, 73-76, 78-87, 89-91, 102-105, and 107-113, are system claims and computer program product claims reciting limitations similar to those of the method claims discussed above. Therefore, the arguments presented above apply to these claims.

Claims 92-94 and 115-117 have been canceled.

The Examiner rejected claims 59-65, 95-101, and 118-119 under 35 USC 102(e) as being anticipated by Laursen et al. This rejection is respectfully traversed.

Claim 59, which has been amended merely to clarify the subject matter of the invention, recites:

> "A computer-implemented method for filtering a directory having a plurality of records, each record having at least two searchable fields, the method comprising:
> accepting a character sequence comprising at least one character, each character having a value;
> in response to each of at least a subset of the characters, iteratively filtering a display of the directory by:
> for each record, determining whether the character sequence matches the record by:
> comparing the character sequence with at least two fields associated with the record; and
> designating the record as a match if the character sequence matches at least one of the fields associated with the record;
> wherein each field associated with the record comprises one selected from the group consisting of:
> at least one searchable field in the record;
> at least one field derived from at least one field in the record; and
> at least one field generated by combining at least two fields in the record; and

displaying at least a subset of records for which the determination indicates a match."

Claim 65 recites:

"A computer-implemented method for filtering a directory having a plurality of records, each record having at least two searchable fields, the method comprising:
accepting a character sequence comprising at least one character, each character having a value;
filtering a directory based on comparison of the accepted character sequence with at least two searchable fields; and
displaying at least a subset of the filtered directory."

In filtering the directory, the claimed method compares the character sequence with at least two fields associated with the record. For example, an entered character sequence can be compared with both an address field and a name field, so that the filter results include records that match with respect to either of those two fields. The user need not specify which of the fields the entered character sequence is intended to match. Support for this claim element appears in the originally filed specification at, for example, paragraphs 0040 and 0068.

Laursen provides no hint or suggestion of such a method. Although the records in Laursen contain multiple fields, Laursen discusses merely a single index field for each record; searches are done with respect to the index field. See, for example, col. 2, lines 27 to 28 ("... the record including at least one field and indexed by a desired index ...") and col. 4, lines 51 to 60 ("... all of the records in the database have a number of fields ... [A]ny field could be used as the index of record 2000 ..."). Additional fields may be displayed in the search results, as shown in the Figures of

Laursen. But there is no discussion in Laursen of comparing a query sequence with two or more fields in order to implement a filtering operation, as claimed herein.

Claims 60-62 have been canceled.

Claims 63-64 depend from claim 59, and therefore incorporate all of the limitations of amended claim 59. The arguments presented above therefore apply to these claims, which are distinguishable from Laursen for the reasons stated above.

Claims 95, 99-101, and 118-119 are system claims and computer program product claims reciting limitations similar to those of the method claims discussed above. Therefore, the arguments presented above apply to these claims.

Claims 96-98 have been canceled.

The Examiner rejected claims 6-8, 18-20, 41, 55, and 70-72 under 35 USC 103 as being unpatentable over the combination of Grover et al. and Laursen et al. This rejection is respectfully traversed.

With regard to claims 6-8, there is no suggestion in either Grover or Laursen for combining their respective disclosures in the manner proposed by the Examiner. In fact, such a combination is unlikely to be operable or useful. Specifically, since every key in Grover has a numeric value, every character sequence has, at minimum, a numeric interpretation that is in some sense "valid". There is no such thing as an invalid character sequence in Grover. Thus Laursen's display of a "No Match" indicator, as shown in Fig. 3L, would never apply to the keyboard disambiguating technique of Grover. Applicants respectfully submit that the Examiner's proposed com-

bination is therefore nonobvious and unlikely to be useful or practical, and is the product of hindsight reconstruction using Applicants' own disclosure.

Furthermore, even if the references were combined in the manner proposed by the Examiner, such a combination would still fail to yield the claimed invention. Claim 6 recites:

> "The method of claim 1, further comprising:
> responsive to the keystroke sequence not producing a valid result in the first context and in the second context, outputting an invalidity indicator."

Thus, an invalidity indicator is output to inform the user that there is no valid result. This is useful feedback for the user.

Claim 6 contains further limitations by virtue of its dependency relationship with claim 1. The arguments presented above therefore apply to these claims, which are distinguishable from Grover for the reasons stated above. For example, as discussed above, claim 1 recites a method that accepts a keystroke sequence and performs one of two different actions depending on whether or not the sequence produces a valid result in a first context. Specifically, if the sequence does produce a valid result in the first context, it outputs first feedback indicating the keystroke input according to the first context. Alternatively, if the sequence does not produce a valid result in the first context, it performs the following two substeps: first it determines whether the keystroke sequence produces a valid result in a second context, and then, if the keystroke sequence does produce a valid result in the second context,

it outputs second feedback indicating keystroke input according to the second context.

Neither Grover nor Laursen, nor any combination thereof, recites these limitations. Rather, the Examiner's proposed combination would merely add a "No Match" screen (as shown in Fig. 4L of Laursen) to the keyboard disambiguating computer of Grover, which, as discussed above, fails to teach or suggest the specific limitations recited in claim 1 and incorporated into claim 6.

Claims 7-8 depend from claim 6, and therefore incorporate all of the limitations of claim 6. The arguments presented above therefore apply to these claims, which are distinguishable from the combination of Grover and Laursen for the reasons stated above.

Claim 18, which has been amended merely to correct a typographical error, recites:

> " The method of claim 10, further comprising:
> responsive to the keystroke sequence producing a valid result in the first context, performing the directory filtering operation using the accepted keystrokes;
> wherein the directory filtering operation comprises comparing the keystroke sequence with at least two field values in at least one directory record."

Thus, when the keystroke sequence produces a valid result in the first context, the directory filtering operation is performed, which includes comparing the keystroke sequence with at least two field values. For example, an entered keystroke sequence can be compared with both an address field and a name field, so that the fil-

ter results include records that match with respect to either of those two fields. The user need not specify which of the fields the entered character sequence is intended to match.

The Examiner correctly stated that Grover fails to teach such a step. Furthermore, Laursen also provides no hint or suggestion of such a method. Although the records in Laursen contain multiple fields, Laursen discusses merely a single index field for each record; searches are done with respect to the index field. See, for example, col. 2, lines 27 to 28 ("... the record including at least one field and indexed by a desired index ...") and col. 4, lines 51 to 60 ("... all of the records in the database have a number of fields ... [A]ny field could be used as the index of record 2000 ..."). Additional fields may be displayed in the search results, as shown in the Figures of Laursen. But there is no discussion in Laursen of comparing a query sequence with two or more field values in order to implement a filtering operation, as claimed herein.

Claims 18-20 depend from claim 10, which in turn depends from claim 1. Therefore, claims 18-20 incorporate all of the limitations of claims 1 and 10, and is distinguishable from the cited references for the reasons given above with respect to claim 1. Specifically, neither of the cited references, taken alone or in any combination, teaches or suggests a method where first feedback corresponding to a first context is shown when there is a valid result in the first context, and second feedback corresponding to a second context is shown responsive to there being no valid result in the first context, as claimed herein.

With regard to claim 41, there is no suggestion in either Grover or Laursen for combining their respective disclosures in the manner proposed by the Examiner. In fact, such a combination is unlikely to be operable or useful. Specifically, since every key in Grover has a numeric value, every character sequence has, at minimum, a numeric interpretation that is in some sense "valid". There is no such thing as an invalid character sequence in Grover. Thus Laursen's display of a "No Match" indicator, as shown in Fig. 3L, would never apply to the keyboard disambiguating technique of Grover. Applicants respectfully submit that the Examiner's proposed combination is therefore nonobvious and unlikely to be useful or practical, and is the product of hindsight reconstruction using Applicants' own disclosure.

Furthermore, neither of the cited references, taken alone or in any combination, teaches or suggests the claimed invention. Claim 41 recites:

> " The method of claim 37, further comprising:
> responsive to at least one of the accepted keystrokes being invalid in one of the contexts, deleting feedback indicating keystroke input according to said one of the contexts."

Claim 41 depends from claim 37, and therefore incorporates all of the limitations of amended claim 37. As described above in connection with claim 37, if the user is entering keystrokes that are valid in two contexts, feedback corresponding to both contexts is displayed. If the user then enters one or more keystrokes that are invalid in one of the contexts, feedback for that context is deleted from the display. For example, if the user enters a keystroke sequence that can be interpreted numerically and alphabetically, both the numeric and alphabetic feedback are displayed; if the

user then enters a keystroke that has no valid numeric value, the previously displayed numeric feedback is deleted from the display.

In addition, depending on whether the keystroke sequence produces a valid result in the first context or in the second context, a first or second action is performed using the first or second value for each keystroke. Thus, when a keystroke sequence produces a valid result in the first context, first feedback is output and a first action is performed. Conversely, when a keystroke sequence produces a valid result in the second context, second feedback is output and a second action is performed. The user need not explicitly specify the action to be performed, as the claimed method infers the action to be taken based on which valid results exist and in which context. Rather, the invention automatically selects an action to perform depending on which context produces a valid result. For example, if the keystroke sequence produces a valid result in a directory lookup context, the invention filters the directory using the first value for each keystroke; conversely, if the keystroke sequence produces a valid result in a direct-dialing context, the invention direct-dials using the second value for each keystroke. These actions are in addition to outputting feedback indicating keystroke input.

Neither Grover nor Laursen, taken alone or in any combination, teaches or suggests any such technique. There is no mention in either reference of any actions that are taken, in addition to outputting feedback, in response to whether or not valid results exist in different contexts.

Claim 55 depends from claim 52, and incorporates all of the limitations of claim 52. Specifically, as discussed above, the method determines whether all values can be converted to valid numeric values and outputs first feedback in response to a positive determination.

Neither of the references discloses such a step. As stated above, in Grover's device every character entry key includes a numeric value. Thus, it would make no sense to determine whether a series of entered keystrokes all have numeric values, since every keystroke sequence entered via the input method of Grover would satisfy this condition. In other words, Grover does not teach any mechanism determining whether or not the user intends to enter a numeric sequence, since every character entry key has a numeric value.

Laursen also fails to each any step of determining whether a series of key-strokes all have numeric values and outputting appropriate feedback responsive to the determination. Rather, Laursen merely describes displaying records based on query strings entered by the user.

Accordingly, claim 55 is submitted to be patentably distinct from Lauren and Grover, taken along or in any combination.

Claims 70-72 are system claims reciting limitations similar to those of claims 6-8. Accordingly, the arguments presented above for claims 6-8 apply to claims 70-72.

The Examiner rejected claims 26, 49, 67-68, 77, 88, 106, and 114 under 35 USC 103 as being unpatentable over the combination of Grover et al. and Shechter et al. This rejection is respectfully traversed.

Claim 26 depends from claim 25, which in turn depends from claim 24, which in turn depends from claim 1. Claim 26 thus incorporates all of the limitations of claims 1, 24, and 25. Claim 26 further recites that "the direct entry operation comprises a telephone number direct entry operation."

The specific limitations of claim 26, including those incorporated from claims 1, 24, and 25, are not taught or suggested by Grover or Shechter, taken alone or in any combination.

As described above, Grover merely describes techniques for implementing a reduced keyboard wherein a plurality of letters and symbols are assigned to most of the keys. When more than one word matches an entered keystroke sequence, Grover presents a menu from which the user selects the desired word. For example, as shown in Fig. 7g of Grover, where the user has entered a three-stroke sequence, the display shows three possible matching choices in an on-screen menu; the user then manually selects the desired word from the menu.

Shechter merely describes a technique for comparing an entered data string (entered, for example, using a telephone keypad) against a plurality of prestored words to find a closest match to the data string.

Nowhere in either reference is there any mention of a method where first feedback corresponding to a first context is shown when there is a valid result in the

first context, and second feedback corresponding to a second context is shown responsive to there being no valid result in the first context, as recited in claim 1 and incorporated as a limitation in claim 26.

Claim 49 depends from claim 48, which in turn depends from claim 37. Claim 49 thus incorporates all of the limitations of claims 37 and 48. Claim 49 further recites that "the direct entry operation comprises a telephone number direct entry operation."

As discussed above, Grover does not teach or suggest any steps for automatically selecting an action to perform depending on which context produces a valid result. Rather, as acknowledged by the Examiner, Grover merely describes displaying different words that match different interpretations of the keystroke sequence. There is no mention of any actions that are taken, in addition to outputting feedback, in response to whether or not valid results exist in different contexts.

Shechter merely describes a technique for comparing an entered data string (entered, for example, using a telephone keypad) against a plurality of prestored words to find a closest match to the data string.

Nowhere in either reference is there any mention of a method where an action is performed depending on which context produces a valid result, as recited in claim 37 and incorporated as a limitation in claim 49.

Claims 67 and 68 depend from claim 66, and incorporate all of the limitations of claim 66, which is a system claim that contains limitations similar to those of method claim 1. Claims 67 and 68 are thus distinguishable from Grover et al. for the reasons discussed above in connection with claim 1.

Claim 67 further recites, in part:

> "a directory lookup engine, coupled to the string handler, for, responsive to the keystroke sequence producing a valid result in the first context, retrieving a telephone number from a directory record identified by the first value for each keystroke; and
> a dialer, coupled to the directory lookup engine, for, responsive to the keystroke sequence producing a valid result in the first context, dialing the retrieved telephone number."

The claimed system retrieves a telephone number from a directory record and dials it, responsive to the keystroke sequence producing a valid result in a first context. This simplifies the user experience of identifying and dialing telephone numbers. An advantage of the claimed system is that the user need not explicitly specify whether to dial using a directory lookup or using direct entry of a telephone number.

Claim 68 further recites, in part:

> "a dialer, coupled to the string handler, for, responsive to the keystroke sequence not producing a valid result in the first context and producing a valid result in the second context, dialing a telephone number specified by the second value for each keystroke."

The claimed system dials a telephone number specified by the second value for each keystroke, responsive to the keystroke sequence not producing a valid result in a first and producing a valid result in a second context. This simplifies the user experience of identifying and dialing telephone numbers. An advantage of the

claimed system is that the user need not explicitly specify whether to dial using a directory lookup or using direct entry of a telephone number.

Neither Grover nor Shechter, taken alone or in any combination, discloses the claimed invention. The Examiner states, correctly, that Grover fails to teach the claimed limitations.

Shechter merely describes a technique for comparing an entered data string (entered, for example, using a telephone keypad) against a plurality of prestored words to find a closest match to the data string. The portion of Schecter cited by the Examiner, page 4, paragraph 0116, is not applicable to the present invention because it describes an embodiment where the user must set a mode switch to specify which character is intended (rather than automatically determining the user's intent. The exact quote from this section states that "the characters are divided into groups and wherein the reduced keyboard has a mode switch for switching the input keys between each group of characters."

Furthermore, to the extent Schecter describes lookup of telephone numbers, this lookup is performed in response to a user's explicit instruction that he or she is seeking a telephone number (see paragraph 0178, line 2). By contrast, the directory lookup engine of claim 67 retrieves a telephone number responsive to the keystroke sequence producing a valid result in the first context. There is no hint or suggestion in Schecter of performing a directory lookup responsive to such a determination of validity.

Furthermore, there is no hint or suggestion in Schecter of dialing a retrieved number responsive to a valid result, as claimed herein. Contrary to the Examiner's assertion, the only mention in Schecter of any form of "dialing" is in paragraph 0178, which states that the user "dials the call center". Schecter fails to teach any technique of dialing a retrieved number, and does not contain any hint of a telephone dialer for performing such an operation as claimed herein.

Claims 77, 88, 106, and 114 are system claims and computer program product claims reciting limitations similar to those of claims 26 and 49. Accordingly, the arguments presented above for claims 26 and 49 apply to claims 77, 88, 106, and 114.

Claims 120-147 have been added to more particularly specify the subject matter that Applicants regard as their invention. Specifically, claims 120-147 recite the use of a QWERTY keyboard in connection with the claimed invention. These claims depend from other claims discussed above, and are therefore distinguishable from the cited art for the reasons discussed above. Furthermore, Applicants note that none of the cited references uses a QWERTY keyboard. Support for the added claims may be found, for example, at paragraphs 0054-0057 and Fig. 2 of the originally filed specification.

On the basis of the above amendments, consideration of this application and the early allowance of all claims herein are requested.

Should the Examiner wish to discuss the above amendments and remarks, or if the Examiner believes that for any reason direct contact with Applicants' represen-

tative would help to advance the prosecution of this case to finality, the Examiner is invited to telephone the undersigned at the number given below.

Respectfully submitted,
Jeffrey C. Hawkins, Thomas B. Bridgwater, Robert Y. Haitani, Srikiran Prasad and William B. Rees

Dated: SEP 1/04

By: _____

Amir H. Raubvogel
Reg. No. 37,070
Fenwick & West LLP
801 California Street
Mountain View, CA 94306
Phone: (650) 335-7276
Fax: (650) 938-5200